JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 27, No. 1, January–February 2004

Optimization of Interplanetary Solar Sailcraft Trajectories Using Evolutionary Neurocontrol

Bernd Dachwald*
DLR, German Aerospace Center, 51147 Cologne, Germany

Solar sailcraft provide a wide range of opportunities for high-energy low-cost missions. As for all low-thrust spacecraft, finding optimal trajectories is a difficult and time-consuming task that involves a lot of experience and expert knowledge because the convergence behavior of optimizers that are based on numerical optimal control methods depends strongly on an adequate initial guess, which is often hard to find. Even if the optimizer converges to an optimal trajectory, this trajectory is typically close to the initial guess that is rarely close to the global optimum. Artificial neural networks in combination with evolutionary algorithms can be applied successfully for optimal solar sail steering. Because these evolutionary neurocontrollers explore the trajectory search space more exhaustively than a human expert can do by using traditional optimal control methods, they are able to find sail steering strategies that generate better trajectories that are closer to the global optimum. Results are presented for a near Earth asteroid rendezvous mission, a Mercury rendezvous mission, and a Pluto flyby mission, which are then compared with previous results found in the literature.

Nomenclature

A	=	sail area
a_c	=	characteristic acceleration [maximum solar sail acceleration at 1 astronomical unit (AU)]
C_3	=	hyperbolic excess energy
$\boldsymbol{c}$	=	steering law weight vector
$\boldsymbol{e}_r, \boldsymbol{e}_t$	=	radial and transversal unit vector
F	=	fitness function
$\boldsymbol{F}_r, \boldsymbol{F}_t$	=	radial and transversal force component
$\boldsymbol{F}_{\text{SRP}}$	=	solar radiation pressure force
$\boldsymbol{f}$	=	thrust unit vector
J	=	cost function
ℓ	=	number of artificial neural network (ANN) neuron layers
m	=	number of internal ANN parameters
n	=	number of pure local steering laws
$\boldsymbol{n}$	=	sail normal vector
n_i	=	number of neurons in the ith ANN neuron layer
P_0	=	mean solar radiation pressure at 1 AU, (4.563 μN/m^2)
q	=	evolutionary algorithm (EA) population size
r	=	sun-sail distance
$\boldsymbol{r}$	=	position of sailcraft
$\boldsymbol{r}_T$	=	position of target body
$\dot{\boldsymbol{r}}$	=	velocity of sailcraft
$\dot{\boldsymbol{r}}_T$	=	velocity of target body
t	=	time
t_f	=	final time
t_0	=	initial time
$\boldsymbol{X}$	=	ANN input vector
$\mathcal{X}$	=	set of ANN input vectors
$\boldsymbol{x}$	=	state of sailcraft, $(\boldsymbol{r}, \dot{\boldsymbol{r}})$
$\boldsymbol{x}_T$	=	state of target body, $(\boldsymbol{r}_T, \dot{\boldsymbol{r}}_T)$
$\boldsymbol{Y}$	=	ANN output vector
$\mathcal{Y}$	=	set of ANN output vectors
α	=	sail clock angle
β	=	sail cone angle
Δr	=	distance between solar sailcraft and target body at final time
ΔV	=	velocity increment
Δv	=	relative velocity between solar sailcraft and target body at final time
η	=	overall sail efficiency factor
$\boldsymbol{\lambda}$	=	adjoint vector of Lagrange multipliers (costate vector)
$\boldsymbol{\xi}$	=	EA chromosome/individual/string
π	=	internal ANN parameter
$\boldsymbol{\pi}$	=	vector of internal ANN parameters
$\boldsymbol{\Upsilon}$	=	sail steering strategy
Φ	=	ANN network function
$\boldsymbol{1}$	=	vector with 1 elements, for example, $\boldsymbol{1} \in \mathbb{R}^3 = (1, 1, 1)$
$\square$	=	arbitrary variable
$\square^\star$	=	optimal $\square$
$\square(t)$	=	value of $\square$ at time t
$\square[t]$	=	time history of $\square$

Introduction

BECAUSE it utilizes solely the freely available solar radiation pressure for propulsion, solar sailcraft provide a wide range of opportunities for low-cost interplanetary missions, many of which are difficult or impossible for any other type of conventional spacecraft due to their large ΔV requirements. Many of those high-energy missions are of great scientific relevance, such as missions to Mercury and to near Earth objects (NEOs) (asteroids and short-period comets) with highly inclined or retrograde orbits.[1,2] (More than 55% of the NEO population have inclinations larger than 10 deg, and more than 30% have inclinations larger than 20 deg.) Within the inner solar system (including the main asteroid belt) solar sailcraft are especially suited for multiple rendezvous missions and for sample-return missions due to their (at least in principle) unlimited ΔV capability. Even missions to the outer solar system may be enhanced by using solar sails, despite the fact that the solar radiation pressure decreases with the square of the sun–sail distance. For such missions, solar sailcraft may gain a large amount of energy when first approaching the sun closely, thereby performing a so-called solar photonic assist maneuver that turns the trajectory into a hyperbolic one.[3–5] Such trajectories allow reasonable transfer times to the outer planets (and to near interstellar space) without the need to perform any gravity assist maneuver; however, without the use of additional propulsive devices and/or an aerocapture maneuver at the target body, only fast flybys can be achieved due to the associated large hyperbolic excess velocities.

Received 21 November 2002; revision received 14 July 2003; accepted for publication 14 July 2003. Copyright © 2003 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/04 $10.00 in correspondence with the CCC.

*Research Engineer, Institute of Space Simulation, Linder Hoehe; bernd.dachwald@dlr.de. Member AIAA.

Traditionally, solar sailcraft trajectories are optimized by the application of numerical optimal control methods that are based on the calculus of variations. The convergence behavior of these optimizers depends strongly on an adequate initial guess, which is needed before optimization. Therefore, depending on the difficulty and complexity of the problem, finding an optimal solar sailcraft trajectory usually turns into a time-consuming task that involves a lot of experience and expert knowledge in control theory and in astrodynamics. Even if convergence is achieved by the optimizer, the optimal trajectory is typically close to the initial guess that is usually far from the (unknown) global optimum.

Using artificial neural networks (ANNs) in combination with evolutionary algorithms (EAs) as evolutionary neurocontrollers, a novel method for solar sailcraft trajectory optimization is presented that does not depend on an initial guess and runs without the involvement of a trajectory optimization expert. This method is not limited to interplanetary solar sailcraft trajectories, but can be adapted for other trajectory optimization problems, for example, for different propulsion systems, for planetocentric motion, etc.

Orbital Dynamics of Solar Sailcraft

The magnitude and direction of the solar radiation pressure (SRP) force $\boldsymbol{F}_{\mathrm{SRP}}$ acting on a flat and perfectly reflecting solar sail (ideal sail) due to the momentum transfer from the solar photons is completely characterized by the sun–sail distance r and the sail attitude, which is generally expressed by the sail normal vector $\boldsymbol{n}$, whose direction is, according to Fig. 1, usually described by the sail clock angle α and the sail cone angle β.

Figure 2 gives a picture of the forces exerted on an ideal sail of area A by the solar radiation pressure $P = P_0 \cdot (1\ \mathrm{AU}/r)^2$ acting on the sail's center of surface. According to the geometry of Fig. 2,

$$\boldsymbol{F}_{\mathrm{SRP}} = \boldsymbol{F}_r + \boldsymbol{F}_{r'} = 2PA\cos^2\beta\boldsymbol{n} \qquad (1)$$

Thus, in the case of perfect reflection, the SRP force that is acting on the sail is always along $\boldsymbol{n}$. Equation (1) may also be expressed in terms of the sailcraft's characteristic acceleration a_c that is defined as the maximum acceleration at 1 AU: $a_c = 2P_0A/m$,

$$\boldsymbol{F}_{\mathrm{SRP}} = ma_c(1\ \mathrm{AU}/r)^2\cos^2\beta\boldsymbol{n} \qquad (2)$$

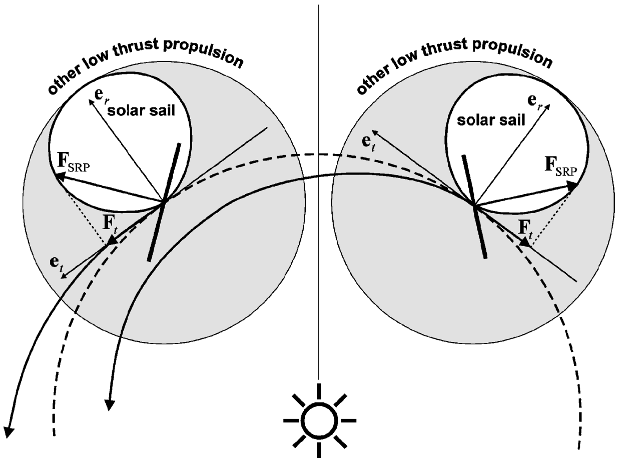
other low thrust propulsion
other low thrust propulsion
solar sail
solar sail
$\mathbf{e}_r$
$\mathbf{e}_r$
$\mathbf{e}_t$
$\mathbf{e}_t$
$\mathbf{F}_{\mathrm{SRP}}$
$\mathbf{F}_{\mathrm{SRP}}$
$\mathbf{F}_t$
$\mathbf{F}_t$

Fig. 3 Spiralling inward, right, and outward, left.

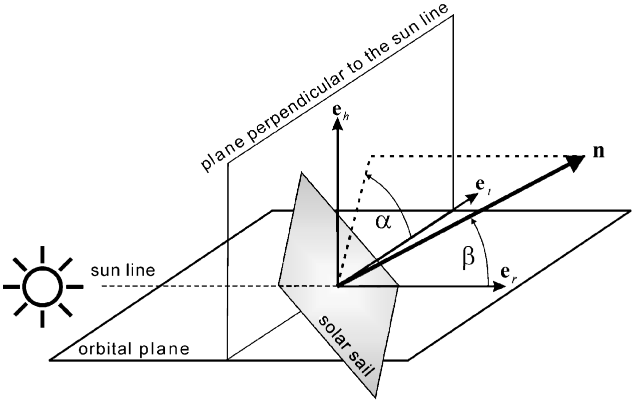

Fig. 1 Definition of the sail clock angle α and the sail cone angle β.

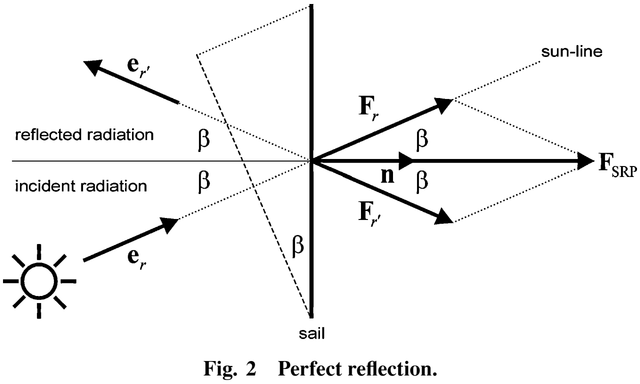

Fig. 2 Perfect reflection.

In the most simple SRP force model for a real solar sail, an overall sail efficiency parameter $\eta < 1$ is used to take into account the nonperfect reflectivity of the sail: $\boldsymbol{F}_{\mathrm{SRP}} = 2\eta PA\cos^2\beta\boldsymbol{n}$. This results in a lower $a_c = 2\eta P_0A/m$ but leaves Eq. 2 unaltered.

The orbital dynamics of solar sailcraft is in many respects similar to the orbital dynamics of other spacecraft, where a small continuous thrust is applied to modify the spacecraft's orbit over an extended period of time. Nevertheless, another continuous thrust spacecraft may orient its thrust vector in any desired direction and vary its thrust level within a wide range, whereas the thrust vector of solar sailcraft is constrained by Eq. (1) to lie on the surface of the $\cos^2\beta$ bubble that is always directed away from the sun (Fig. 3). Nevertheless, by controlling the sail orientation relative to the sun, solar sailcraft can gain orbital angular momentum (if $\boldsymbol{F}_{\mathrm{SRP}} \cdot \boldsymbol{e}_t > 0$) and spiral outward, away from the sun, or lose orbital angular momentum (if $\boldsymbol{F}_{\mathrm{SRP}} \cdot \boldsymbol{e}_t < 0$) and spiral inward, toward the sun.

Simulation Model and Problem Statement

Because a real sail is neither flat nor a perfect reflector, a thorough trajectory analysis must take into account the optical properties of the real sail, which are also time varying due to the erosive effects of the space environment. Nevertheless, for preliminary trajectory analysis, as done here, some simplifications may be made.

1) The sailcraft is moving under the sole influence of solar gravitation and radiation. The sun is a point mass and a point light source. Forces from celestial bodies other than the sun are neglected. Also neglected are disturbing forces, which are much smaller than the sun's gravitational force and the SRP force, for example, caused by the solar wind and the aberration of light.

2) The nonperfect reflectivity of the sail is described by an overall sail efficiency factor η.

3) The sail attitude can be changed instantaneously.

This paper deals with the problem of finding the optimal interplanetary solar sailcraft trajectory (rendezvous or flyby) to a given target body. In terms of optimal control theory the rendezvous problem and the flyby problem can be stated as follows.[6]

The rendezvous problem (formulation 1) is to find a sail normal vector (control vector) history $\boldsymbol{n}[t]$ (for $t_0 \le t \le t_f$) that forces the state $\boldsymbol{x}(t) = (\mathbf{r}(t), \dot{\boldsymbol{r}}(t))$ of the sailcraft from its initial value $\boldsymbol{x}(t_0)$ to the state $\boldsymbol{x}_T(t)$ of the target body [thus, obeying the terminal constraint $\boldsymbol{x}(t_f) = \boldsymbol{x}_T(t_f)$] and, at the same time, minimizes the cost function

$$J = \int_{t_0}^{t_f} \mathrm{d}t = t_f - t_0$$

[Because solar sailcraft do not consume any propellant, only the transfer time is minimized and, unlike for other spacecraft, the final mass $m(t_f)$ is not a part of the cost function.] The resulting state history $\boldsymbol{x}^\star[t]$ is the optimal trajectory for the given rendezvous problem.

The flyby problem (formulation 1) is to find a sail normal vector history $\boldsymbol{n}[t]$ (for $t_0 \leq t \leq t_f$) that forces the position $\boldsymbol{r}(t)$ of the sailcraft from its initial value $\boldsymbol{r}(t_0)$ to the position $\boldsymbol{r}_T(t)$ of the target body [thus, obeying the terminal constraint $\boldsymbol{r}(t_f) = \boldsymbol{r}_T(t_f)$] and, at the same time, minimizes the cost function

$$J = \int_{t_0}^{t_f} \mathrm{d}t = t_f - t_0$$

The resulting state history $\boldsymbol{x}^\star[t]$ is the optimal trajectory for the given flyby problem.

Thus, both trajectory optimization problems are actually problems of finding the optimal control vector history $\boldsymbol{n}^\star[t]$.

Traditional Trajectory Optimization

Traditionally, solar sailcraft trajectories are optimized by the application of numerical optimal control methods that are based on the calculus of variations. These methods can be divided into direct methods, such as nonlinear programming (NLP) methods, and indirect methods, such as neighboring extremal methods and gradient methods. Before optimization, the NLP methods and the gradient methods require an initial guess for the control vector history $\boldsymbol{n}[t]$, whereas the neighboring extremal methods require an initial guess for the starting adjoint vector of Lagrange multipliers $\boldsymbol{\lambda}(t_0)$ (costate vector).[6] The convergence behavior of all of those methods is very sensitive to the respective initial guess,[3,7] so that trajectory optimization becomes sometimes more art than science.[7] If convergence is achieved, a local optimum is found that is typically close to the initial guess and far from the global optimum. If convergence could not be achieved, a new initial guess has to be conceived. Because similar initial guesses often produce very dissimilar optimization results, the initial guess can not be improved iteratively, and the search for a good trajectory can turn into a very time-consuming task.

ANNs

Insofar as they are inspired by the processing of information in animal nervous systems, ANNs are a computability paradigm that is alternative to conventional serial digital computers. ANNs are massively parallel, analog, fault tolerant, and adaptive.[8] They are composed of processing elements (called neurons) that model the most elementary functions of the biological neuron. Linked together, those elements show some characteristics of the brain, for example, learning from experience, generalizing from previous examples to new ones, and extracting essential characteristics from inputs containing noisy and/or irrelevant data, so that they are relatively insensitive to minor variations in its input to produce consistent output.[9]

Because the neurons can be connected in many ways, ANNs exist in a wide variety. Here, however, only feedforward ANNs are considered. Typically, feedforward ANNs have a layered topology, where the neurons are organized hierarchically in a number ℓ of so-called neuron layers. The first neuron layer is called the input layer and has n_1 input neurons that receive the network's input. The last neuron layer is called the output layer and has n_ℓ output neurons that provide the network's output. All intermediate layers/neurons are called hidden layers/neurons. [An example for a layered feedforward ANN with three input neurons, one hidden layer with two hidden neurons, and one output neuron is shown in the Appendix.] If a sigmoid activation function for the neurons is used (see the Appendix), a layered feedforward ANN can be regarded as a continuous parameterized function (called network function)

$$\Phi_\pi : \mathcal{X} \subseteq \mathbb{R}^{n_1} \to \mathcal{Y} \subset \mathbb{R}^{n_\ell}$$

that maps from a set of inputs $\mathcal{X}$ onto a set of outputs $\mathcal{Y}$. The parameter vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)$ of the network function comprises the m internal parameters of the ANN (the weights and the biases of the neurons, see the Appendix).

If the correct output is known for a set of given inputs (the training set), the network error, that is, the difference between the actual output and the correct output, can be measured and utilized to learn the optimal network function $\Phi^\star :=\Phi_{\pi^\star}$ by adapting $\boldsymbol{\pi}$ in a way that the network error is minimized. For this kind of learning problems, a variety of learning algorithms have been developed to determine the optimal network parameter vector $\boldsymbol{\pi}^\star$, the backpropagation algorithm, a gradient-based method, being the most widely known.[9,10]

Reinforcement Learning and Neurocontrol

Learning algorithms for ANNs that rely on a training set fail when the correct output for a given input is not known. This is the case for so-called reinforcement learning (RL) problems, where the optimal behavior of the learning system (called agent) has to be learned solely through interaction with the environment that gives an immediate or delayed scalar evaluation (reinforcement) of the agent's behavior.[11,12] The optimal behavior of the agent is defined as the one that maximizes the sum of positive reinforcements and minimizes the sum of negative reinforcements over time. Delayed reinforcement learning (DRL) problems commonly arise in the optimal control of dynamical systems.[11]

Operating within so-called neurocontrollers (NCs), ANNs have been successfully applied to this class of learning problems.[9] Neurocontrol approaches to solve RL problems can be divided into two categories, indirect (or critic-based) ones and direct ones.[13] The direct neurocontrol approach, which is used here, employs a single ANN, which is called the action model (or action network). The action network controls the dynamic system by providing a control vector $\boldsymbol{Y}(t) \in \mathcal{Y}$ from some input vector $\boldsymbol{X}(t) \in \mathcal{X}$ that contains the information that is relevant to perform the control task (system state, environmental state, etc.). The more commonly used indirect critic-based neurocontrol approach, which is not used here, employs additionally a system model and a second ANN, which is called evaluation model or critic network. Based on the system model, the evaluation model provides a prediction of the evaluation of the action that is considered by the action model.[9,13,14] Henceforth, to keep things simple, the term NC is used for the ANN that is, precisely speaking, the action network of the NC.

NCs can also be applied to the optimal control problem of solar sailcraft trajectory optimization, which is a DRL problem: If an NC is used to direct the sailcraft's trajectory by controlling the sail attitude $\boldsymbol{n}(t)$, then this NC receives a single reinforcement for its control vector history $\boldsymbol{n}[t]$, that is, for its behavior, at the final time t_f, when the trajectory can be evaluated. Note that the NC's behavior is completely characterized by its network function Φ_π (that is again completely characterized by its parameter vector $\boldsymbol{\pi}$). The next section will address a learning method for DRL problems that may be used for determining the NC's optimal network function $\Phi_{\pi^\star}$.

EAs and Evolutionary Neurocontrol

EAs, sometimes called genetic algorithms (GAs), are proven to be robust methods for finding global optima in very high-dimensional search spaces. They have been successfully applied as a learning method for ANNs,[15–17] as well as for a wide range of other optimization problems. Therefore, they are also expected to be an efficient method for finding the NC's optimal network function.

EAs use a vocabulary borrowed from biology. The key element of an EA is a population that comprises numerous individuals $\boldsymbol{\xi}_k (k \in \{1, \ldots, q\})$, which are potential solutions to the given optimization problem. All individuals of the (initially randomly created) population are evaluated according to a fitness function F (analogous to a cost function) for their suitability to solve the problem. Their allocated fitness value $F(\boldsymbol{\xi}_k)$ is crucial for their probability to reproduce and to create offspring into a newly created population because a selection scheme (the environment) selects fitter individuals with a greater probability for reproduction than less fit ones. The selected parents undergo a series of genetic transformations (mutation, recombination) to produce offspring that consists of a mixture of the parent's genetic material. Under the selection pressure of the environment, the individuals, which are also called chromosomes or strings, strive for survival. After some reproduction cycles, the population converges against a single solution $\boldsymbol{\xi}^\star$, which is in the best case the globally optimal solution to the given problem.

The application of an EA to search for the NC's optimal network function makes use of the fact that a NC parameter vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)$ can be mapped onto a real valued string $\boldsymbol{\xi}$ of length m, which provides an equivalent description of the NC's network function. By searching for the fittest individual (string) $\boldsymbol{\xi}^\star$, the EA searches for the NC's optimal network function $\Phi^\star$. Such NCs, which employ an EA for learning, are called evolutionary neurocontrollers (ENCs).

Before applying an NC to the optimization of solar sailcraft trajectories, the NC's input set $\mathcal{X}$ and output set $\mathcal{Y}$ have to be defined adequately, that is, the following questions have to be answered: What does the NC get as input? How do we interpret the NC's output, or rather what is the NC expected to do? This is crucial for the NC's performance on the problem because we can not expect the NC to make something out of nothing. Before providing an answer to those questions, sail steering strategies have to be addressed.

Solar Sail Steering

A pure local steering law (PLSL) may be defined as a steering law that changes (increases, decreases, or adjusts to some given reference value) some actual osculating orbital element of spacecraft with a maximum rate. For obtaining PLSLs, we may use Lagrange's planetary equations in Gauss form (see Ref. 18) because these equations describe the rate of change of a body's osculating elements due to some (disturbing and/or propulsive) acceleration. If we have defined n PLSLs, each PLSL $i \in \{1, \ldots, n\}$ gives a direction, along which the SRP force has to be maximized. This direction may be expressed by a thrust unit vector $\boldsymbol{f}_i$. From $\boldsymbol{f}_i$, the related sail normal vector $\boldsymbol{n}_{f_i}$ (and, thus, the sail clock angle α_{f_i} and the sail cone angle β_{f_i}) can be calculated.

To change more than one orbital element at the same time, the n PLSLs can be blended. For that reason, a vector $\boldsymbol{c} \in [0, 1]^n$ of weight factors (called steering law weight vector) may be defined in a way that each $\boldsymbol{c}$ defines a blended local steering law (BLSL) by giving the (blended) thrust unit vector

$$\boldsymbol{f} = \frac{\sum_{i=1}^{n} c_i \boldsymbol{f}_i}{\left|\sum_{i=1}^{n} c_i \boldsymbol{f}_i\right|} \tag{3}$$

from the (pure) thrust unit vectors $\boldsymbol{f}_i$. Again, the related sail normal vector $\boldsymbol{n}_f$, sail clock angle α_f, and sail cone angle β_f can be calculated from $\boldsymbol{f}$.

A sail steering strategy may now be defined as some function $\boldsymbol{\Upsilon} : \mathcal{X} \to [0, 1]^n$, which gives the actual steering law weight vector $\boldsymbol{c}(t) \in [0, 1]^n$ from some vector of input variables $\boldsymbol{X}(t) \in \mathcal{X}$. The rendezvous and the flyby problem may then be reformulated.

The rendezvous problem (formulation 2) is to find a sail steering strategy $\boldsymbol{\Upsilon} : \mathcal{X} \to [0, 1]^n$ (for $t_0 \le t \le t_f$) that forces the state $\boldsymbol{x}(t)$ of the sailcraft from its initial value $\boldsymbol{x}(t_0)$ to the state $\boldsymbol{x}_T(t)$ of the target body [thus, obeying the terminal constraint $\boldsymbol{x}(t_f) = \boldsymbol{x}_T(t_f)$] and, at the same time, minimizes the cost function $J = t_f - t_0$. The resulting steering strategy $\boldsymbol{\Upsilon}^\star$ is the optimal steering strategy for the given rendezvous problem.

The flyby problem (formulation 2) is to find a sail steering strategy $\boldsymbol{\Upsilon} : \mathcal{X} \to [0, 1]^n$ (for $t_0 \le t \le t_f$) that forces the position $\boldsymbol{r}(t)$ of the sailcraft from its initial value $\boldsymbol{r}(t_0)$ to the position $\boldsymbol{r}_T(t)$ of the target body [thus, obeying the terminal constraint $\boldsymbol{r}(t_f) = \boldsymbol{r}_T(t_f)$] and, at the same time, minimizes the cost function $J = t_f - t_0$. The resulting steering strategy $\boldsymbol{\Upsilon}^\star$ is the optimal steering strategy for the given flyby problem.

Thus, both trajectory optimization problems are actually problems of finding the optimal sail steering strategy $\boldsymbol{\Upsilon}^\star$.

To use PLSLs and BLSLs is just one method for obtaining sail steering strategies. Because those steering strategies have implicit knowledge about how to change the orbital elements in an optimal way, they can be considered as indirect steering strategies. However, the implementation of steering strategies is also possible without the use of local steering laws, for example, by providing the thrust unit vector $\boldsymbol{f}$ directly. Because such steering strategies do not have implicit knowledge about what orbital elements are and how they can be changed, they can be considered as direct steering strategies.

Solar Sailcraft Trajectory Optimization Using ENC

For the implementation of sail steering strategies, as defined earlier, an ENC may be used. In this case, each NC's parameter vector $\boldsymbol{\pi}$ defines a steering strategy $\boldsymbol{\Upsilon}_\pi : \mathcal{X} \to \mathcal{Y}$, and the EA is used to determine the optimal NC parameter vector $\boldsymbol{\pi}^\star$ that results, after some transformations according to Fig. 4, in the optimal sailcraft trajectory $\boldsymbol{x}^\star[t]$.

Two different output sets $\mathcal{Y}$ have been considered, one representing an indirect steering strategy and one representing a direct one.

1) The NC provides the steering law weight vector $\boldsymbol{c}$, $\boldsymbol{\Upsilon} : \mathcal{X} \to \{\boldsymbol{c}\}$ (indirect steering strategy).

2) The NC provides the thrust unit vector $\boldsymbol{f}$, along which the SRP force has to be maximized, $\boldsymbol{\Upsilon} : \mathcal{X} \to \{\boldsymbol{f}\}$ (direct steering strategy). Note that the thrust unit vector $\boldsymbol{f}$ can be calculated from the NC output $\boldsymbol{Y} \in (0, 1)^3$ via the simple transformation $\boldsymbol{f} = (2\boldsymbol{Y} - \mathbf{1})/|2\boldsymbol{Y} - \mathbf{1}|$ that maps from $(0, 1)^3$ onto $(-1, 1)^3$.

Because it is reasonable to assume for a robust steering strategy that the actual optimal SRP force direction $\boldsymbol{n}(t)$ depends at any time t on the actual sailcraft state $\boldsymbol{x}(t)$ and the target body $\boldsymbol{x}_T(t)$, $\mathcal{X} = \{(\boldsymbol{x}, \boldsymbol{x}_T)\}$ was used for the domain of solar sail steering strategies, thus, $\boldsymbol{\Upsilon} : \{(\boldsymbol{x}, \boldsymbol{x}_T)\} \to \mathcal{Y}$. (Note that a steering strategy that is defined in this way does not depend explicitly on time.)

Now the final picture of solar sail steering using an ENC can be drawn (Fig. 5): To find the optimal trajectory, the ENC method is run in two loops. Within the (inner) trajectory integration loop, a NC steers the sailcraft to fly the trajectory that is completely defined by the NC's parameters, which are set by the EA in the (outer) NC optimization loop.

For solar sail steering, the NC takes the actual sailcraft state $\boldsymbol{x}(t)$ and that of the target body $\boldsymbol{x}_T(t)$ as input values and maps from

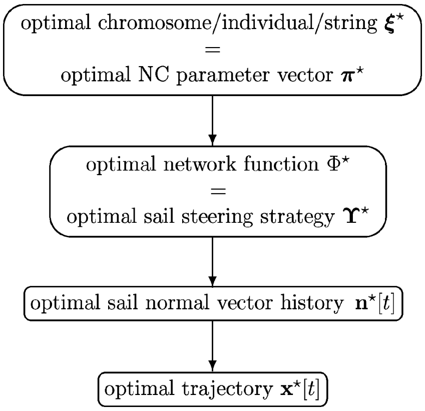
optimal chromosome/individual/string $\boldsymbol{\xi}^\star$
=
optimal NC parameter vector $\boldsymbol{\pi}^\star$
optimal network function $\Phi^\star$
=
optimal sail steering strategy $\boldsymbol{\Upsilon}^\star$
optimal sail normal vector history $\mathbf{n}^\star[t]$
optimal trajectory $\mathbf{x}^\star[t]$

Fig. 4 Transformation of the optimal chromosome $\xi^\star$ into the optimal trajectory $x^\star[t]$.

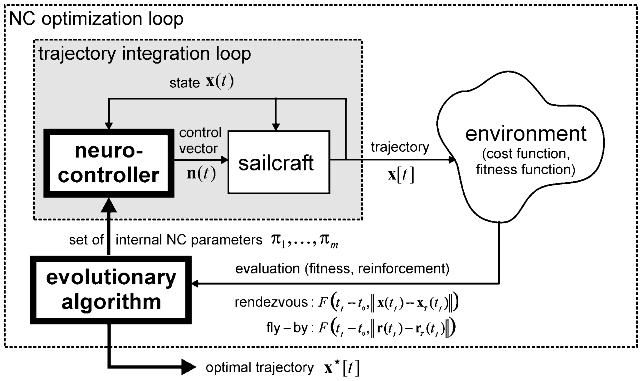
NC optimization loop
trajectory integration loop
state $\mathbf{x}(t)$
neuro-controller
control vector
$\mathbf{n}(t)$
sailcraft
trajectory
$\mathbf{x}[t]$
environment
(cost function, fitness function)
set of internal NC parameters $\pi_1, \ldots, \pi_m$
evolutionary algorithm
evaluation (fitness, reinforcement)
rendezvous : $F\left(t_f - t_0, \left\|\mathbf{x}(t_f) - \mathbf{x}_T(t_f)\right\|\right)$
fly-by : $F\left(t_f - t_0, \left\|\mathbf{r}(t_f) - \mathbf{r}_T(t_f)\right\|\right)$
optimal trajectory $\mathbf{x}^\star[t]$

Fig. 5 Trajectory optimization using an evolutionary neurocontroller.

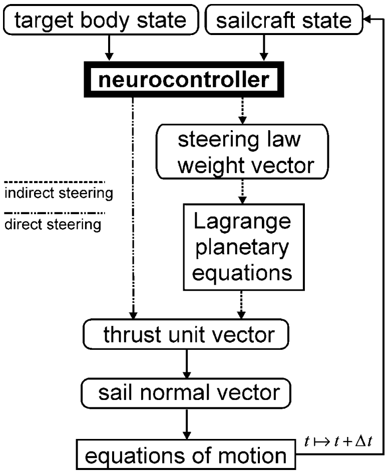

Fig. 6 Solar sail steering using a neurocontroller (trajectory integration loop).

them onto some output values (Fig. 6). If the NC implements an indirect steering strategy, its n output values are interpreted as the required steering law weight vector $\boldsymbol{c}(t)$. When this steering law weight vector is used, the required thrust unit vector $\boldsymbol{f}(t)$ can be calculated from Langrange's planetary equations. If the NC implements a direct steering strategy, its three output values are directly interpreted as the required thrust unit vector $\boldsymbol{f}(t)$. Thus, the required sail normal vector $\boldsymbol{n}_f(t)$ [or $\alpha_f(t)$ and $\beta_f(t)$] can be calculated in both cases. Having done this, the sail normal vector is inserted into the equations of motion, and they are integrated over a time period Δt to get $\boldsymbol{x}(t + \Delta t)$. This state is fed back into the NC. The trajectory integration loop stops when the terminal constraint is practically satisfied [$\|\boldsymbol{x}(t) - \boldsymbol{x}_T(t)\| \leq \epsilon$ for the rendezvous problem or $\|\boldsymbol{r}(t) - \boldsymbol{r}_T(t)\| \leq \epsilon$ for the fly-by problem] or if some time limit is reached. Then, in the NC optimization loop, the NC's parameter vector, that is, its trajectory, is rated by the EA's fitness function. Because we can not expect the NC to generate only trajectories that obey the final constraint, the respective constraint has to be included into the fitness function, so that $F = F[t_f - t_0, \|\boldsymbol{x}(t_f) - \boldsymbol{x}_T(t_f)\|]$ for the rendezvous problem and $F = F[t_f - t_0, \|\boldsymbol{r}(t_f) - \boldsymbol{r}_T(t_f)\|]$ for the flyby problem. As already mentioned, this fitness is crucial for the probability to reproduce and to create offspring. Under this selection pressure, the ENC generates more and more suitable trajectories. The ENC finally converges against a single steering strategy, which gives in the best case a near globally optimal trajectory for the rendezvous problem (near globally, because global optimality can rarely be proven except by complete enumeration, which is not feasible).

Results

The ENC method described earlier was applied to a variety of solar sailcraft trajectory optimization problems. In the sequel, the results for three mission examples are presented, for which expert-generated trajectories are available.[3,4,19–22] Their recalculation using ENC (with the ENC representing a direct steering strategy) reveals that, for all three mission examples, the trajectories found with traditional optimization methods are quite far from the global optimum.

Rendezvous with 1996FG$_3$

The first mission example (Fig. 7) is a rendezvous with near Earth Asteroid (NEA) 1996FG$_3$, a mission that will not be too demanding for moderate performance solar sailcraft of the first generation ($a_c = 0.14$ mm/s^2). The ENC trajectory for the 1996FG$_3$ rendezvous mission is 123 days (7.5%) faster (for the same launch date) than the conventionally generated trajectory[21,22] ($\Delta r < 50{,}000$ km and $\Delta v < 100$ m/s at time of rendezvous). (This accuracy is compatible with the accuracy of the expert generated trajectory; nevertheless, using ENC, it can be improved further, $\Delta r < 0.04$ km and $\Delta v < 0.04$ m/s, at the cost of 10 days additional flight time.) The ENC trajectory also reduces the C_3 requirement from $C_3 = 4$ km^2/s^2 to $C_3 = 0$ km^2/s^2, which eventually allows the sailcraft to be launched with a smaller and, thus, cheaper launcher.

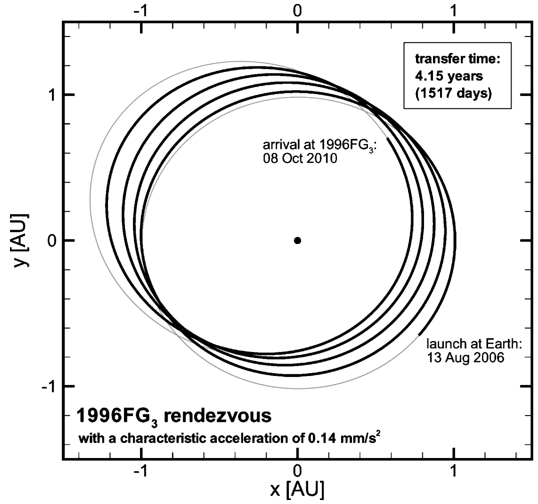

Fig. 7 NEA 1996FG$_3$ rendezvous trajectory.

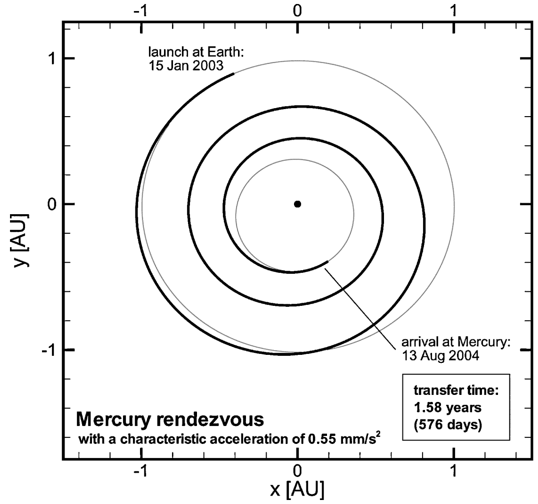

Fig. 8 Mercury rendezvous trajectory.

Rendezvous with Mercury

The second mission example (Fig. 8) is a Mercury rendezvous with a more advanced solar sailcraft ($a_c = 0.55$ mm/s^2). Here, the ENC trajectory is 89 days (13.4%) faster (for the same launch date) than the conventionally generated trajectory[3,19,20] ($\Delta r < 250$ km and $\Delta v < 10$ m/s at time of rendezvous), both with $C_3 = 0$ km^2/s^2.

Pluto Flyby

The third mission example (Figs. 9 and 10) is a Pluto flyby with a quite advanced solar sailcraft ($a_c = 1.0$ mm/s^2). Here, the ENC trajectory is 837.5 days (17.4%) faster (for the same launch date, $\Delta r < 52{,}000$ km and $\Delta v = 21.1$ km/s at time of flyby) than the conventionally generated trajectory,[3,4] both with $C_3 = 0$ km^2/s^2. The minimum solar distance was set to 0.49 AU because this was also the minimum solar distance of the expert-generated trajectory. The large improvement in flight time is because the ENC achieved a double solar photonic assist, whereas the expert-generated trajectory performs only a single solar photonic assist. During the first solar photonic assist, the eccentricity of the orbit is increased to raise the aphelion of the orbit. This highly elliptic orbit is advantageous for the second solar photonic assist because it yields a large velocity at perihelion, so that the solar sail can attain a large final velocity increment. Thus, the double solar photonic assist allows the solar sailcraft to gain more energy during the last solar photonic

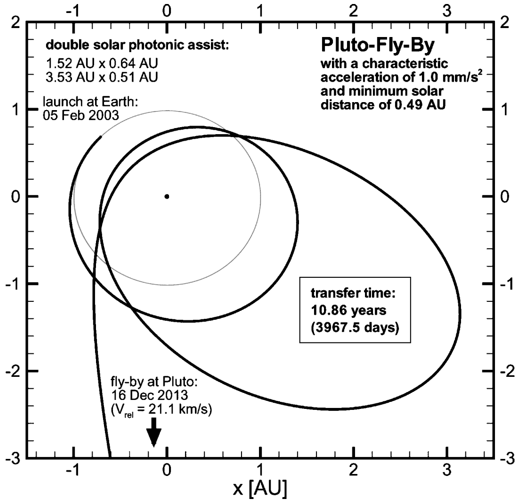

Fig. 9 Pluto flyby trajectory using a double solar photonic assist.

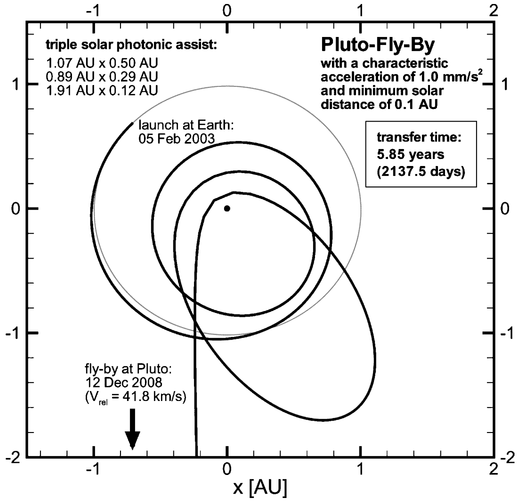

Fig. 10 Pluto flyby trajectory using a triple solar photonic assist.

assist that turns the trajectory hyperbolic. The additional time spent in the inner solar system is more than compensated by the larger solar system escape velocity. Along the trajectory, the maximum temperature of the sail would not exceed 365 K if the sail was Al coated (for high reflectivity, ≈88%) on the front side and Cr coated (for high emissivity, ≈55%) on the back side. This is well below the temperature limit of conventional polyimide (PI) substrates, like Kapton® ($T_{\mathrm{PI,max}} \approx 570$ K) (Ref. 3).

If the solar sail is allowed to approach the sun more closely, a triple solar photonic is also possible. This reduces the flight time to Pluto even further (2667.5 days, $\Delta r < 410$ km and $\Delta v = 41.8$ km/s at time of flyby); however, the thermal load on the sail would be significantly higher ($T_{\max} \approx 730$ K) for such a close solar approach, exceeding the temperature limit of conventional PI substrates.

ENC Parameters and Constraints

To perform trajectory optimization with an ENC, the following parameters have to be fixed: the NC's input set, the NC's output set, the NC's topology (number of hidden neuron layers and number of hidden neurons), some EA parameters (population size, mutation rate, selection scheme, etc.), and the EA's fitness function.

Various combinations of those parameters have been investigated. The performance of the ENC was found to be relatively robust with respect to different settings of most of the parameters. It depends, however, strongly on the choice for the EA's fitness function. This is reasonable because this function has not only to decide autonomously which trajectories are good and which are not, but also which trajectories are promising for future cultivation and which are not. For most problems, direct and indirect steering strategies produce similar results. Apart from the results, direct steering strategies are, however, more elegant.

The typically used NCs had a single hidden layer with about 30 neurons. The maximum number of integration steps was usually set to values between 200 and 1000, allowing the NC to change the sail attitude every 1–10 days. Depending on the number of integration steps, the total computation time for one trajectory optimization run was in the order of a few hours on a 1.3-GHz personal computer, during which the EA reproduced and tested between 10,000 and 100,000 trajectories.

The ENC-generated trajectories are quite accurate with respect to the terminal constraint; however, they are not optimal solutions in the strict sense because the terminal constraint is not exactly met. To improve the accuracy of the trajectories further, an ENC trajectory can be taken as the initial guess for a direct numerical optimal control method such as NLP.

Conclusions

The results shown indicate clearly that the novel method of using an ENC for solar sail steering is a very promising approach for finding near globally optimal solar sailcraft trajectories. The obtained trajectories are fairly accurate with respect to the terminal constraint. If a more accurate trajectory is required, the ENC solution can be used as an initial guess for traditional trajectory optimization methods. Evolutionary neurocontrol may be applied to a wide variety of low-thrust trajectory optimization problems, including different propulsion systems and planetocentric trajectories.

Nevertheless, before ENCs could be considered as a versatile and robust tool for generating near globally optimal trajectories by someone without basic knowledge in control theory and astrodynamics, further research on convergence, stability, and robust parameter settings should be done.

Appendix: Layered Feedforward ANNs

ANNs can be divided into feedforward ANNs and recurrent ANNs according to the connectivity of the neurons. An ANN is a feedforward one if there exists a numbering method that numbers all neurons in a way that there is no connection from a neuron with a number i to a neuron with a number $j < i$. An ANN is a recurrent one if such a numbering method does not exist.

Typically, feedforward ANNs have a layered topology, where the set $\mathcal{N}$ of neurons is divided into ℓ subsets $\mathcal{N}_1, \ldots, \mathcal{N}_\ell$ (called neuron layers) so that only connections from $\mathcal{N}_{k-1}$ go to $\mathcal{N}_k$ for all $k \in \{2, \ldots, \ell\}$. $\mathcal{N}_1$ is called the input layer and has n_1 input neurons, which receive the network's input $\boldsymbol{X} \in \mathbb{R}^{n_1}$. $\mathcal{N}_\ell$ is called the output layer and has n_ℓ output neurons, which provide the network's output $\boldsymbol{Y} \in (0, 1)^{n_\ell}$. All other layers/neurons are called hidden layers/neurons (if $\ell > 2$). As an example, Fig. A1 shows an ANN with $\ell = 3$

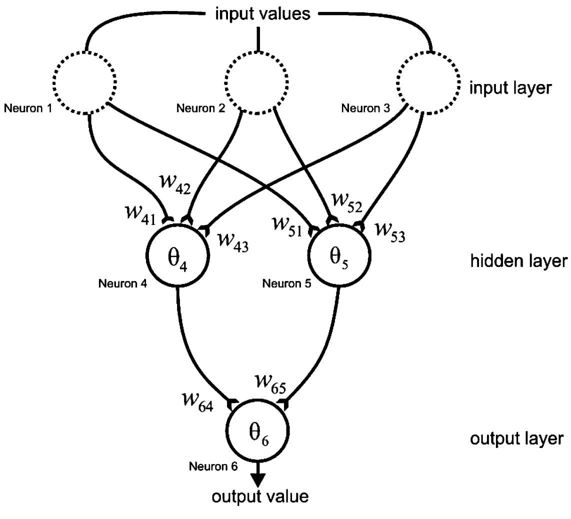

Fig. A1 Layered feedforward ANN.

layers, $n_1 = 3$ input neurons, one hidden layer with $n_2 = 2$ hidden neurons, and $n_3 = 1$ output neuron (3–2–1 network).

Each neuron $i \in \mathcal{N}_k$ $(k \in \{2, \ldots, \ell\})$ has a so-called activation function that maps from the neuron's input value(s) onto a single output value. The most commonly used activation function for feedforward networks is the sigmoid $s_\gamma : \mathbb{R} \to (0, 1)$, defined by

$$s_\gamma(x) = 1/(1 + e^{-\gamma x}) \tag{A1}$$

where the constant γ defines the slope of the function.

When the sigmoid activation function s_γ is used, a layered feedforward ANN can be described as a directed graph in which each node (neuron) i in a layer $\mathcal{N}_k$, $k \in \{2, \ldots, \ell\}$, performs the function

$$y_i = 1 \Bigg/ \left\{ 1 + \exp\left[-\gamma \left(\sum_j w_{ij} x_j - \theta_i \right) \right] \right\} \tag{A2}$$

where $y_i \in (0, 1)$ is the output of neuron i, the $x_j \in (0, 1)$ are the output values of the neurons j in the previous layer $\mathcal{N}_{k-1}$, the $w_{ij} \in \mathbb{R}$ are the connection weights between the neurons j and neuron i, and $\theta_i \in \mathbb{R}$ is the so-called bias (or threshold) of neuron i. Each neuron i in the input layer $\mathcal{N}_1$ gives directly one component of the network's input values $X_i \in \mathbb{R}$:

$$y_i = X_i \tag{A3}$$

Layered feedforward ANNs with a sigmoid activation function for the neurons can be regarded as a continuous parameterized function $\Phi_{\pi_1,\ldots,\pi_m} : \mathbb{R}^{n_1} \to (0, 1)^{n_\ell}$, called the network function, where the m parameters of the network function $\pi_1, \ldots, \pi_m$ are the connection weights w_{ij} and the biases θ_i of the neurons. For the ANN in Fig. A1, we have $m = 11$, $\pi_1 = \theta_4$, $\pi_2 = w_{41}$, and $\pi_3 = w_{42}, \ldots, \pi_{11} = \theta_6$.

Using Kolmogorov's theorem, it can be proven that any continuous function can be represented exactly by a finite network of computing units, though the general learning problem of determining the values for a given network's parameters is NP complete (see Ref. 10). In simple terms, this means that it is very improbable that an algorithm exists that is able to solve the problem in finite time (within the age of the universe) if the number of unknown variables gets large, though a guessed solution can be checked in finite time. In most practical cases, however, no exact function representation is demanded, but a finite approximation error is accepted for the network function, so that an approximate solution for the problem can be found in reasonable time.

References

[1] "Near Earth Objects—Dynamic Site" [online database], URL: http://newton.dm.unipi.it/cgi-bin/neodys/neoibo [cited 8 July 2003].

[2] Dachwald, B., Seboldt, W., and Häusler, B., "Performance Requirements for Near-Term Interplanetary Solar Sailcraft Missions," AAAF-S29-5, Association Aeronautique and Astronautique of France, France, May 2002.

[3] Leipold, M., "Solar Sail Mission Design," Ph.D. Dissertation, Lehrstuhl für Flugmechanik und Flugregelung, Technische Universität München, Munich, Germany, 1999, pp. 21–22, 76–117.

[4] Leipold, M., "To the Sun and Pluto with Solar Sails and Micro-Sciencecraft," *Acta Astronautica*, Vol. 45, No. 4, 1999, pp. 549–555.

[5] Sauer, C. G., "Optimum Solar-Sail Interplanetary Trajectories," AIAA Paper 76-792, Aug. 1976.

[6] Stengel, R. F., *Optimal Control and Estimation*, Dover, New York, 1994, pp. 185–192, 254–270.

[7] Hartmann, J. W., "Low-Thrust Trajectory Optimization Using Stochastic Optimization Techniques," M.S. Thesis, Dept. of Aeronautical and Astronautical Engineering, Univ. of Illinois at Urbana–Champaign, IL, 1996, pp. 41–56.

[8] Rojas, R., "Was können neuronale Netze?," *Mathematische Aspekte der angewandten Informatik*, Wissenschaftsverlag, Mannheim, Germany, 1994, pp. 55–88.

[9] Dracopoulos, D. C., *Evolutionary Learning Algorithms for Neural Adaptive Control*, Springer, London, 1997, pp. 47–70, 97–109.

[10] Rojas, R., *Neural Networks. A Systematic Introduction*, Springer, Berlin, 1996, pp. 149–182, 263–285.

[11] Keerthi, S. S., Ravindran, B., "A Tutorial Survey of Reinforcement Learning," Dept. of Computer Science and Automation, Indian Inst. of Science, Bangalore, India, 1995, pp. 1–4.

[12] Sutton, R. S., and Barto, A. G., *Reinforcement Learning*, MIT Press, Cambridge, MA, 1998, pp. 3–23.

[13] Likas, A., and Lagaris, I., "Training Reinforcement Neurocontrollers Using the Polytope Algorithm," *Neural Processing Letters*, Vol. 9, No. 2, 1999, pp. 119–127.

[14] Barto, A. G., "Connectionist Learning for Control," *Neural Networks for Control*, MIT Press, Cambridge, MA, 1990, pp. 5–58.

[15] Whitley, D., Dominic, S., Das, R., and Anderson, C. W., "Genetic Reinforcement Learning for Neurocontrol Problems," *Machine Learning*, Vol. 13, No. 2, 1993, pp. 259–284.

[16] Tsinas, L., and Dachwald, B., "A Combined Neural and Genetic Learning Algorithm," *Proceedings of the First IEEE Conference on Evolutionary Computation*, Vol. 2, Inst. of Electrical and Electronics Engineers, New York, 1994, pp. 770–774.

[17] Yao, X., "Evolutionary Artificial Neural Networks," *Encyclopedia of Computer Science and Technology*, Vol. 33, Marcel Dekker, New York, 1995, pp. 137–170.

[18] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, revised ed., AIAA Education Series, AIAA, Reston, VA, 1999, pp. 484–490.

[19] Leipold, M., Seboldt, W., Lingner, S., Borg, E., Herrmann, A., Pabsch, A., Wagner, O., and Brückner, J., "Mercury Sun-Synchronous Polar Orbiter with a Solar Sail," *Acta Astronautica*, Vol. 39, No. 1, 1996, pp. 143–151.

[20] Leipold, M., Pfeiffer, E., Groepper, P., Eiden, M., Seboldt, W., Herbeck, L., and Unkenbold, W., "Solar Sail Technology for Advanced Space Science Missions," 52nd International Astronautical Congress, International Astronautical Federation Paper 01-S.6.10, Oct. 2001.

[21] Seboldt, W., Leipold, M., Rezazad, M., Herbeck, L., Unkenbold, W., Kassing, D., and Eiden, M., "Ground-Based Demonstration of Solar Sail Technology," 51st International Astronautical Congress, International Astronautical Federation Paper 00-S.6.11, Oct. 2000.

[22] Jessberger, E. K., Seboldt, W., Glassmeier, K.-H., Neukum, G., Pätzold, M., Arnold, G., Auster, H.-U., deNiem, D., Guckenbiehl, F., Häusler, B., Hahn, G., Hanowski, N., Harris, A., Hirsch, H., Kührt, E., Leipold, M., Lorenz, E., Michaelis, H., Möhlmann, D., Mottola, S., Neuhaus, D., Palme, H., Rauer, H., Rezazad, M., Richter, L., Stöffler, D., Willnecker, R., Brückner, J., Klingelhöfer, and G., Spohn, T., "ENEAS—Exploration of Near-Earth Asteroids with a Sailcraft," Proposal for a Small Satellite Mission within the Space Sciences Program of the German Aerospace Center, DLR, German Aerospace, Research Center, Muenster, Germany, 2000, pp. 27–30.